Optima

Verlaine, Bertrand; Jureta, Ivan J.; Faulkner, Stéphane

Published in:
Service Oriented Computing and Applications (SOCA)

DOI:
10.1007/s11761-013-0151-5

Publication date:
2014

Document Version
Peer reviewed version

Service Oriented Computing and applications manuscript No.
(will be inserted by the editor)

Optima: a domain-specific model for prioritization and conflicts management in requirements engineering for services intermediaries

Bertrand Verlaine · Ivan J. Jureta · Stéphane Faulkner

Received: date / Accepted: date

Abstract New business models are set up thanks to Web technologies. In this work, we focus on services intermediary companies. They generate value through the (automatic) selection of third-party services and the (automatic) delivery of the combinations of these services to consumers. Such companies face the problem of deciding which services to select and deliver in order to maximize their profit.
The two main paper objectives are (i) to design the generic business model of services intermediaries, and (ii) to propose an optimization model. The latter enables to choose the consumer requirements that will be satisfied in order to maximize profit. This model ranks implementable solutions based on various financial aspects. They are related to cost and revenue information that is associated with the requirements. It can support the decision-making process that aims at selecting a profit-maximizing set of requirements for services intermediaries' system-to-be. Indeed, the proposed model solves the conflicts between requirements and prioritizes the optional requirements. We argue for the relevance of the optimization model via an example and simulations.

Keywords Requirements Engineering · Service-oriented Systems · Services Intermediary · Requirements Prioritization · Conflicts Resolution

B. Verlaine · I. J. Jureta · S. Faulkner
PReCISE Research Center – University of Namur
8 Rempart de la Vierge, B-5000 Namur, Belgium
Tel.: +32 81 725316
E-mail: bertrand.verlaine@fundp.ac.be

I. J. Jureta
E-mail: ivan.jureta@fundp.ac.be

S. Faulkner
E-mail: stephane.faulkner@fundp.ac.be

1 Introduction

Following the use of Information and Communication Technologies (ICT) in many organizations, most of supply chains are undergoing a reorganization of their processes. The intermediary is one of the actors active in many supply chains. It is involved in the reorganization of many business processes. Its three main functions are (i) to organize the meeting between buyers and sellers, (ii) to facilitate the processes supporting the business transactions and (iii) to provide or to improve the mechanisms underlying these transactions in order to respect laws, specific regulations or ethical rules [2, 16]. The current re-intermediation of supply chains is mainly based on Web technologies. They automate the core processes of intermediaries that is selecting and reselling goods and services. In this context, such an intermediary can be called an *e-intermediary*.
Along with this re-intermediation of business processes, researchers in information systems (IS) are developing a new paradigm: the *Service-oriented Computing* (SOC). From a business point of view, SOC is "a set of flexible [Web] services and processes that a business wants to expose to its customers, partners, or internally to other parts of the organization, and the same [Web] services can be recombined and supplemented to support changes to or an evolution of business models and requirements over time" [5]. Web Services (WS) are pieces of software which execute well-delimited tasks once they are contacted. Their significant advantage lies in the standardization of the technologies used to support their implementation. As a result, more and more providers propose their goods and services through WS technologies.

Considering these two facts, a new kind of intermediary is developing: *the services intermediary*. The latter buys and resells services using WS technologies. It also has to engineer and develop an IS to support its main business functions. The first step of this software engineering process is

the Requirements Engineering (RE). A major issue in RE is decision-making during which the two following questions must be answered: *How to resolve conflicts between requirements?* and *What are the optional requirements to implement?*. Seeing that services intermediaries have specific ISs for supporting their core business, we have to investigate how to adapt or create solutions for resolving conflicts between requirements and prioritizing optional requirements.
Contributions. First, this paper defines the concept of services intermediary and discusses the main causes of its rapid expansion. It also proposes a reference business model of services intermediation.
The major contribution is an optimization model called Optima. The latter combines and solves two issues of RE, namely, the resolution of conflicts and the prioritization of requirements. It can be used in the domain of the services intermediation. Based on Techne [22], an abstract requirements modeling and analysis language, it can generate an ordered list of the candidate solutions identified in a goals model. Solutions are sorted according to their respective expected profits. Each satisfies the stakeholders' needs. Note that a running example is also developed along with the optimization model in order to ease its understanding.
Organization. Firstly, we examine the current re-intermediation of supply chains. That discussion enables to develop the business model of the services intermediation (§2). Then, we explain how two main issues in RE (i.e., the prioritization of requirements and the conflicts resolution) have a crucial role during the RE stage of services intermediaries' ISs (§3). In that section, we also introduce the Techne language and we use an example throughout the development of the optimization model (§4). In §5, we present some results emerging from the use of the optimization model. Before concluding, we discuss how stakeholders preferences should be handled when using our solution (§6). Finally, we position our work among other published papers (§7) and conclude it by proposing some future works (§8).

2 Intermediation and service-oriented computing

In this section, we propose the business model of the *services intermediation*. In economic literature, a business model is defined as the strategy followed by a company to create value and generate revenue from its products and services offered for sale [35]. But in computing literature, this term often refers to the representation of the business concepts present in the real world [36]. Within the scope of this work, the economic interpretation is preferred.
In §2.1, we position the notion of e-intermediation with regards to the notions of e-business and e-commerce. This allows to design the business model of services intermediation (§2.2). Its analysis leads to the main research issue tackled in this paper (§2.3).

2.1 E-commerce and e-intermediation

Although the borders of e-business are not very clear in the literature, the following definition is generally accepted: e-business refers to the exchange of information through ICT in order to conduct organization's business processes, both externally and internally [8, 20, 23]. E-commerce is closely linked to e-business and, at least partially, e-commerce principles and techniques are part of the e-business notion. E-commerce can be defined as the use of ICT to support business transactions, i.e., the sales and purchases processes achieved by the organization [8, 20]. Our work focuses on a specific business model which is more and more used in e-commerce: the e-intermediation (concept defined in §2.2).

2.2 The services intermediation

2.2.1 Definition of a services intermediary

Due to the evolution of ICT, value chains tended to drastically reduce the amount of intermediaries in comparison with traditional physical markets [8]. Intermediaries are economic agents that purchase products and services from other economic agents and then resell them by taking a profit margin. This evolution, called *disintermediation*, often ensues in a direct contact between the producers and their final customers. The latter must find, browse and compare the products and services proposed in order to make their purchase decision. However, due to the boom in products and services available on the Internet, e-markets without any intermediaries face significant problems such as the information distortion, lacks of (neutral) comparisons, high providers searching costs and the lack of trust between business partners [8]. This leads to a new trend: the e-intermediation which is an ICT based re-intermediation. It refers to "the introduction into the supply chain of specialist intermediary firms who use electronic commerce technologies to facilitate supply chain performance" [6].

A second significant technological trend lies in the Service-oriented Computing (SOC) based on IT-services[1]. The latter are self-describing and self-contained modular applications designed to execute well-delimited tasks, and that can be described, published, located, and invoked over a network [37].

These two trends gave birth to the business model of *services intermediation*.

[1] In order to differentiate two notions commonly called "service" in their respective domains, we call the concept of "service" in the SOC an IT-*service*, and the business service keeps its short denomination, that is to say *service*. The latter can be defined as "a change in the condition of a person, or a good belonging to some economic entity, brought about as the result of the activity of some other economic entity [. . .]" [18].

A services intermediary is an electronic agent that resells services purchased through SOC technologies after, possibly, their composition.

Services intermediaries are thus close to IT-service brokers: some brokers can assemble several IT-services in order to provide one composite IT-service such as services intermediaries. But it differs from the economic point of view: services intermediaries select the services providers and add an economic value to the resold composite IT-service while IT-service brokers are mainly in charge of communities of services providers [7]. Their main objective is the maintenance of common IT-services registries and not the services trading [40]. Despite this significant difference, Optima could be used by IT-service brokers which would like to develop a business activity based on the composition and resale of IT-services present in their community.

2.2.2 The business model of services intermediation

The Strategic Dependency Business Model [30] derived from the i^* language is used in Fig. 1. Its main purpose is to represent (reference) generic business models. Only the useful modeling concepts for the business model of a services intermediary are illustrated in the legend of Fig. 1 and briefly explained hereinafter.

An Actor is an economic agent involved in some relationships exposed by the business model.

A Resource is a business factor which is required in a relationship by an actor in order to achieve desired outcome.

A Softgoal is a vague objective of an actor concerning the economic relationship he has with another actor.

A Task is the carrying out of an activity resulting from the economic relationship the actors have together.

A Dependency relationship models the dependence of an actor towards another actor concerning the execution of a task, the achievement of a softgoal or the providing of a resource. The nature of the dependum defines the nature of the dependency relationship.

As illustrated in Fig. 1, the services intermediary is in relationship with several services consumers and several IT-services providers through ICT. With regards to the notion of e-commerce, the services intermediary is involved in a many-to-many e-relationship: it is indeed implied in electronic transactions both at sell-side and buy-side. Except the fact that they are electronically based, the kind of relationships between the services consumers and the services intermediary is not relevant in the scope of this work. The latter focuses indeed on the IT-services selection among several providers based on the potential solutions, i.e., the potential (composite) services that the services intermediary can propose to its customers. Therefore, our discussion mainly focuses on business-to-business e-commerce. Understanding how a services intermediary earns money is a key point in this work. As illustrated in Fig. 1, the services intermediary pays for IT-services delivery and is paid by services consumers when it resells them. As a result, the profit made *on the sales* comes from the difference between the payments made to the IT-services providers and the money received from the services consumers. Of course, the services intermediary will face other costs such as investments, running costs, staff costs, and so on.

2.3 Research question

A significant issue remains unsolved for services intermediation companies: How a single services intermediary can decide which are the (composite) services it should propose to its customers? Another way to state this question is: *How should a services intermediary choose services that maximize its profit?* To provide a satisfactory answer to this question related to the software engineering process, we first need to discuss the implications of the RE stage on the business model studied (see §3). This paper tackles that part of the problem. We also have to position this work towards two significant and different periods in software engineering, namely the design time and the runtime. Optima is conceived to suit the design time of an IS. Thus, we assume to build the system-to-be of the services intermediary from scratch.

Fig. 2 gives an overview of how Optima works. On the one hand, a requirements model is designed based on the elicited needs expressed by stakeholders. Very often, requirements models contain conflicts to resolve. Moreover, in many cases, all the optional requirements cannot be implemented because of, e.g., time, money or skill reasons. Therefore, they must be prioritized. That means that the requirements model can lead to multiple potential solutions.
On the other hand, new web-based businesses often need to advertise. One of the best solutions is to buy keywords to search engine companies [26]. Based on the description of the chosen keywords, the number of purchases is estimated for each potential solution identified in the requirements model. Thanks to this information, the revenue received from the implementation of each requirement is estimated.
As a result, the profit obtained from each potential solution is computed by aggregating the revenue and costs associated to each requirement belonging to that solution. That allows to rank the potential solutions from a financial point of view. It can help system engineers and stakeholders to make implementation decisions.

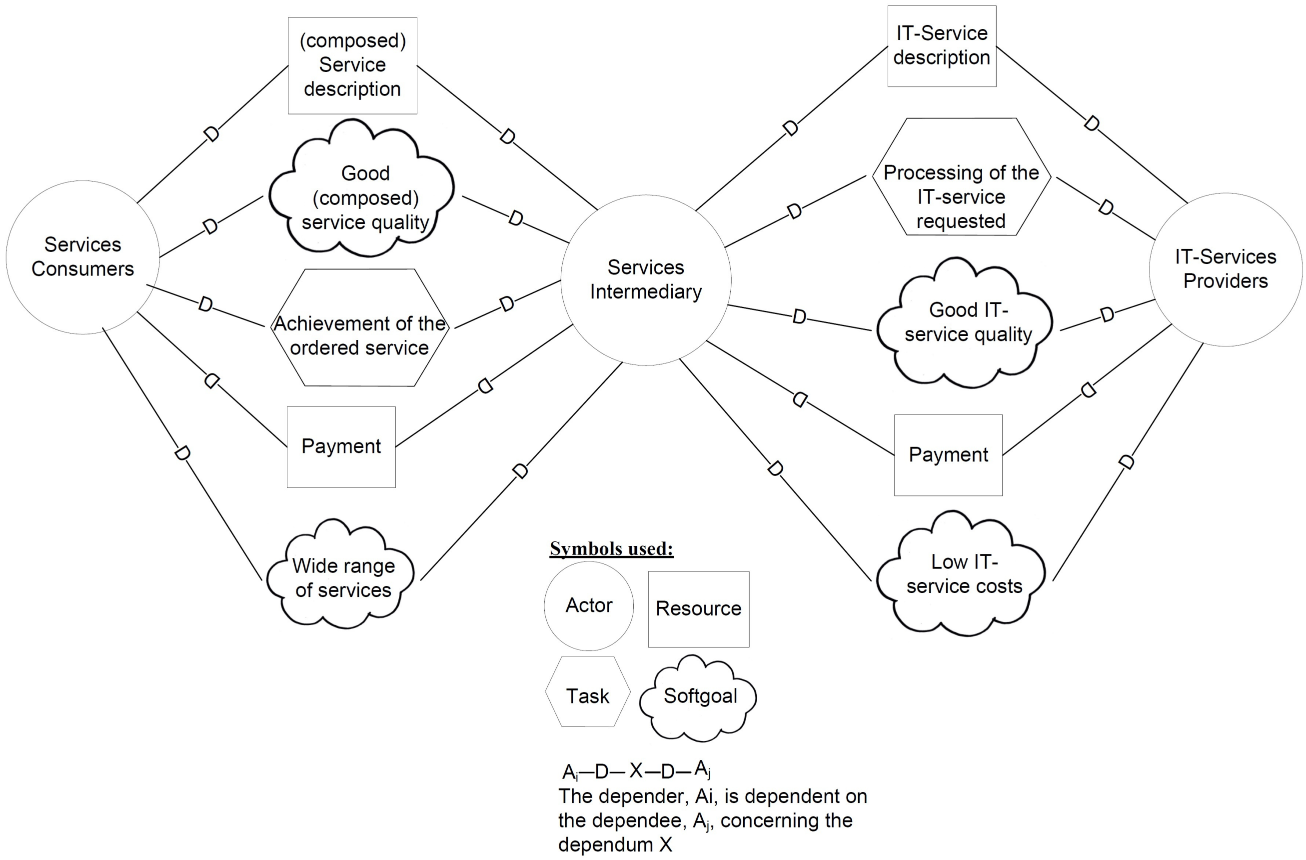


Fig. 1 The representation of the *Services Intermediation* business model using the Strategic Dependency Business Model

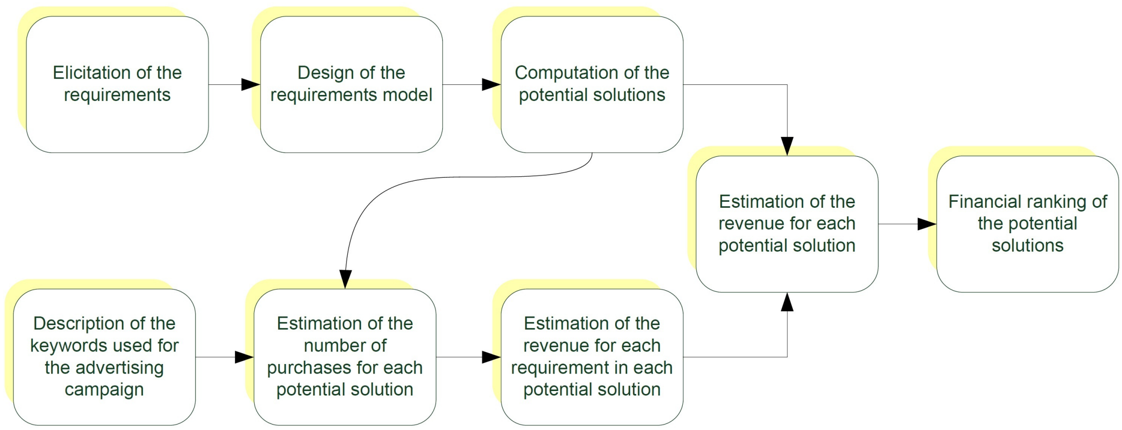


Fig. 2 The use process of Optima in a nutshell

3 Requirements engineering for services intermediaries

3.1 The arbitration stage in requirements engineering

Seeing that services intermediaries are electronic agents, their IS must be engineered like any other IS. RE is commonly defined as the process by which project stakeholders are identified, and by which their requirements are elicited, modeled and analyzed. This process must lead to the specification of the system-to-be.

Once the project stakeholders identified, the requirements can be elicited, refined and modeled. Those requirements must be analyzed during the RE process and validated at the end. During the analysis step, several arbitrations often take place. It consists of the resolution of conflicts and the prioritization of optional requirements by considering, principally, their (estimated) implementation costs, the technological possibilities, the financial consequences on the future revenue and costs and the expressed stakeholders' preferences. A consistent and satisfying solution to the requirements problem should issue from the arbitration stage. A consistent solution is a solution without any conflicts among the selected requirements [22]. A solution is satisfying when it contains all the mandatory requirements as well as potentially optional requirements.

As explained in §2.2, ISs used by services intermediaries are designed and built inside the service-oriented paradigm. That is why we situate our work within research on the Service-oriented Requirements Engineering, or more precisely in this paper, the IT-Service-oriented Requirements Engineering (SORE). The SORE process, like the traditional RE process, aims at analyzing and specifying the requirements and constraints of an IT-service oriented system-to-be [47]. As stated in [3], "there is still no standard Requirements Engineering process defined for Service Oriented Software" although the traditional RE processes cannot be used as it is in the IT-service oriented paradigm [3, 49]. SORE "differs from traditional requirements engineering as it assumes that the concerned application will be developed in a [Service-Oriented Environment] framework running in an SOA [Service-Oriented Architecture] infrastructure" [48]. In this paper, we propose an optimization model (see §4) to maximize the profit of a services intermediary. This model could help to solve the open issue stated above: helping to make design decisions during the requirements arbitration stage leading to the implementation of a services intermediary IS.

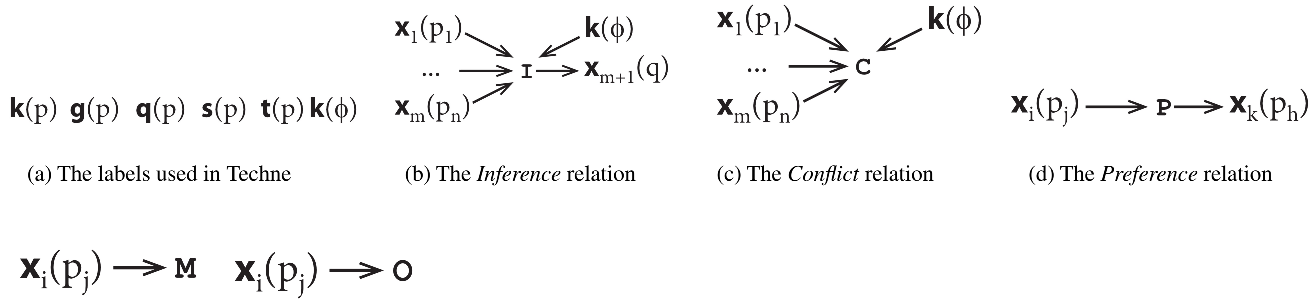
3.2 Techne: an abstract requirements modeling and analysis language

Once the requirements are elicited, it is necessary to reason about them in order to achieve the arbitration work. In this way, Jureta and colleagues propose an abstract Requirements Modeling Language (RML) called Techne [22]. Techne is designed to be used during the early phase of the RE process, which includes the requirements modeling and analysis stages. The work proposed in this paper contributes to solve the third issue of an RML [22]: among candidate solutions of a services intermediary system-to-be, Optima suggests the most profitable one. A candidate solution is a consistent set of mandatory requirements which should satisfy some optional requirements.

Techne enabled to model and structure the different requirements expressed by stakeholders. Depending on their nature, those requirements are classified in goals, quality constraints, softgoals, tasks and assumptions; their labels are respectively **g**(), **q**(), **s**(), **t**() and **k**() as illustrated in Fig. 3(a). The distinction among the expressed requirements is made thanks to a Core Ontology of REquirements (CORE) [21]. Desires expressed by stakeholders are captured by goals –if they refer to a binary and verifiable condition– or by quality constraints –if they refer to a non-binary verifiable condition. Vague and/or non-verifiable desires are captured by softgoals to be then approximated by goal(s) or quality constraint(s). Stakeholders' intentions to act become instances of tasks, which are then achieved either by the system-to-be or by the stakeholders, or both. Beliefs are instances of domain assumptions which state how the system-to-be will perform the tasks in order to satisfy the goals, the quality constraints and, as best as possible, the softgoals. The requirements are referred by statements: let d and u be two atomic statements respectively stating "Generate revenue from an English-speaking travel Web site" and "Highlight the sponsored trips in the services lists proposed to customers". Those two statements are instances of, respectively, a goal denoted as $\mathbf{g}(d)$ and a quality constraint denoted as $\mathbf{q}(u)$. Note that complex statements, used in some requirements relations (see below), are denoted by Greek letters, e.g., $\mathbf{k}(\varphi)$.

Techne allows to express five relations between the captured requirements, which are listed and explained below: Figures 3(b) to 3(f) illustrate them. Note that each **x** in Figures 3(b) to 3(f) are arbitrary labels of Fig. 3(a).

- Inference: The inference relation, illustrated in Fig. 3(b), relates two requirements when the first one (the premise) is an immediate consequence of the second one (the conclusion). An inference relation can have several premises. For each inference relation, a complex statement classified as an assumption abbreviates "if $\mathbf{x}_1(p_1)$ and ... and $\mathbf{x}_m(p_n)$, then $\mathbf{x}_{m+1}(p_{n+1})$".
- Conflict: The conflict relation is illustrated in Techne as in Fig. 3(c). This relation expresses the inconsistencies between requirements. It gives a complex statement, referring to an assumption, which states "if $\mathbf{x}_1(p_1)$ and ... and $\mathbf{x}_m(p_n)$, then contradiction". The consequence is the impossibility of having those conflicting requirements in the same candidate solution because of the inconsistency present in the solution.
- Preference: The preference relation expresses the comparison of two requirements made by stakeholders. In Fig. 3(d), the requirement $\mathbf{x}_i(p_j)$ is preferred to the requirement $\mathbf{x}_k(p_h)$
- Is-mandatory: The is-mandatory relation indicates that a requirement is mandatory in the system-to-be. Each mandatory requirement must be included in every candidate solution. In Fig. 3(e), the requirement $\mathbf{x}_i(p_j)$ is mandatory.
- Is-optional: Unlike the is-mandatory relation, the is-optional relation is used to indicate that a requirement is non-mandatory. That means that a solution can be a candidate solution without including it although its implementation in the system-to-be is viewed as positive by the stakeholders. In Fig. 3(f), the requirement $\mathbf{x}_i(p_j)$ is optional.



(a) The labels used in Techne (b) The *Inference* relation (c) The *Conflict* relation (d) The *Preference* relation (e) The *Is-mandatory* relation (f) The *Is-optional* relation

Fig. 3 Illustration of the allowed grammar in Techne

Section 3.3 proposes an example of an r-net. An r-net is the representation of a requirements model using the Techne constructors about which it is possible to reason and find the candidate solutions. Finding the candidate solutions lies in the identification of the different conflicts-free solutions which include all the mandatory requirements and some optional requirements. Ernst et al. have recently proposed an algorithm automating the identification of the candidate solutions in an r-net [12]. Nevertheless, this can be manually carried out in small r-nets. First, all the mandatory requirements must be identified. Then, conflict-free optional requirements are added to the set of mandatory requirements. Lastly, each conflict in the r-net gives additional candidates solution by choosing one and then the other conflicting requirements. They are added to the set of requirements built at the previous step. All possible combinations of requirements must be considered.

3.3 Description of a services intermediary active in the travel business

Let USAtrip be a services intermediary. This company proposes different travel services to its customers thanks to a Web interface. These travel services are trips to New-York. Its targeted market focuses on European English-speaking customers. By using IT-service technologies, USAtrip will propose and resell travel services on the Web (e.g., flights, car transports, accommodations, and so on) which will be bought from different providers (e.g., hotels, flying companies, cruise shipping, coaches, and so on). USAtrip follows the business model of a services intermediary (see Fig. 1 and related explanations).

Table 1 List of the requirements expressed by the USAtrip stakeholders.

d :	Generate revenue from an English-speaking travel Web site.
p_1 :	Sell travel services to New-York.
p_2 :	Sell single travel services to New-York.
p_3 :	Sell composed travel services to New-York.
p_4 :	Compose and sell atomic travel services.
p_5 :	Sell transport services to New-York.
p_6 :	Sell accommodation services in New-York.
p_7 :	Sell leisure activities in New-York.
u_1 :	Generate revenue from the IT-services providers.
u_2 :	Ask providers to pay for a prioritization of their travel products.
u_3 :	Highlight the sponsored trips in the services lists proposed to customers.
r_1 :	Sell reliable information on travel possibilities.
r_2 :	Charge fees to customers for travel information.
v_1 :	Trips are advised to customers in a neutral way.
v_2 :	Provide opinions, rating and advice about accommodations, leisure activities and transport companies.
v_3 :	Allow customers to provide feedbacks on their trips.
v_4 :	Display all travel offers in the same layout.
w_1 :	The travel Web site is attractive for the potential customers.
w_2 :	Propose more than 250 different accommodations in New-York.
w_3 :	Propose trips from 2 nights to 45 nights.

3.3.1 Modeling of the USAtrip requirements

Fig. 4 shows the r-net structuring the requirements elicited from the USAtrip stakeholders. Table 1 lists the requirements modeled in Fig. 4. The categorization *d,p,u,r,v* and *w* is used to ease the reading of the r-net. If a requirement is mandatory in an r-net, then all its subsequent requirements –they are its premises– are also mandatory unless otherwise stated by an *is-optional* relationship.

3.3.2 Solutions identified in the Techne requirements model

As explained in §3.2, a candidate solution is a consistent set of mandatory requirements which should also satisfy some of the optional requirements.

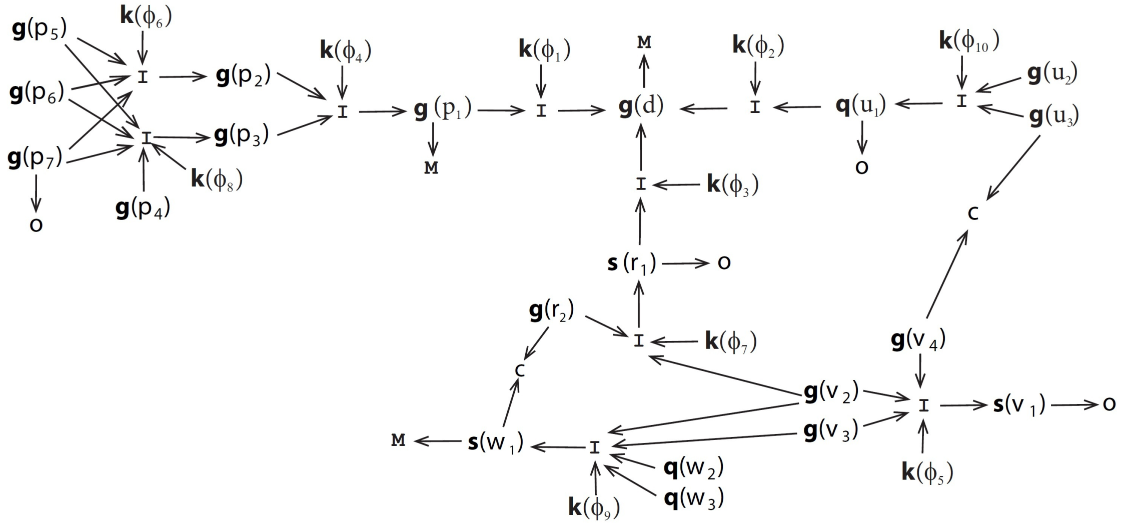


Fig. 4 The r-net of the USAtrip requirements modeled with the Techne RML

Let the solution S_1 be the minimal sub r-net, i.e., the consistent r-net exclusively made up of the mandatory requirements. S_1 contains the following requirements:

$$g(p_4) \wedge g(p_5) \wedge g(p_6) \wedge g(v_2) \wedge g(v_3) \wedge q(w_2) \wedge q(w_3) \quad (1)$$

From the USAtrip r-net illustrated in Fig. 4, we can find the candidate solutions denoted S_2 and S_3. They are detailed in Eqs. 2 and 3. All candidate solutions satisfy the requirements problem: the selected requirements should lead to a satisfying system-to-be in comparison with the stakeholders' needs.

$$S_1 \; along \; with \; the \; requirements \; set \; \{g(p_7) \wedge g(u_2) \wedge g(u_3)\} \; gives \; S_2 \quad (2)$$

$$S_1 \; along \; with \; the \; requirements \; set \; \{g(p_7) \wedge g(v_4)\} \; gives \; S_3 \quad (3)$$

Fig. 5 highlights the solutions S_1, S_2 and S_3. They are based on the initial r-net of the USAtrip's system-to-be.

You can notice than the optional softgoal $s(r_1)$ never belongs to any of the solutions. Indeed, some design choices directly result from the requirements expressed by the stakeholders. A conflict between an optional requirement and a mandatory requirement (or their premises) causes the rejection of the optional requirement in all solutions. Other choices such as choosing between S_2 and S_3 cannot be achieved without a decision tool as the one provided herein.

We extend the current set of solutions –currently including S_1, S_2 and S_3– by progressively removing the optional requirements. The process followed is: for each candidate solution, make a new solution by removing sets of potential solutions. Those sets are composed of, progressively, $1, 2, \ldots, x$ optional requirements. Stop when you obtain the minimal sub r-net. From the solutions S_2 and S_3 (S_1 already being the minimum sub r-net), we find three new potential solutions:

$$S_1 \; along \; with \; the \; requirement \; g(p_7) \; gives \; S_4 \quad (4)$$

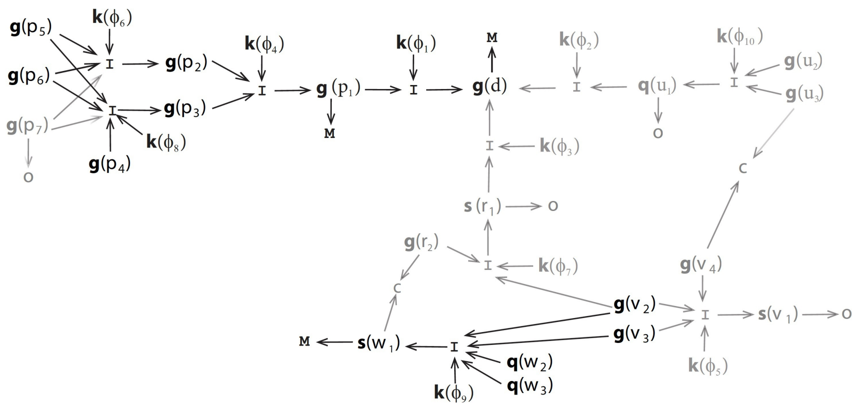
$$S_1 \; along \; with \; the \; requirements \; set \; \{g(u_2) \wedge g(u_3)\} \; gives \; S_5 \quad (5)$$

$$S_1 \; along \; with \; the \; requirement \; g(v_4) \; gives \; S_6 \quad (6)$$

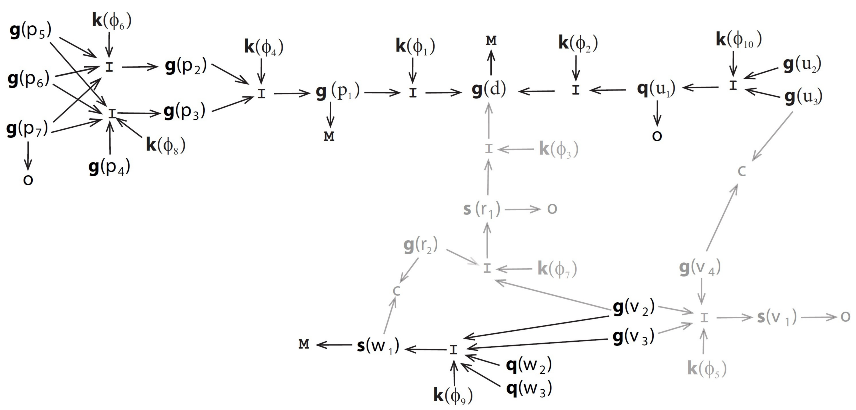
The reason why we remove progressively the optional requirements in order to obtain S_4 to S_6 lies in the final objective of our optimization model. Optima attempts to propose the best possible solution which minimizes the implementation costs while maximizing the profit obtained from the implementation of the services intermediary's system-to-be. Some of the optional requirements could be, for instance, too expensive to implement or too expensive to maintain compared with the revenue gained from them. In the scope of this work, we consider that the implementation costs are the addition of the costs due to the specification of the system-to-be, its coding, its testing and its final installation before its use.

4 Optima: an optimization model for decision making in requirements engineering for services intermediaries

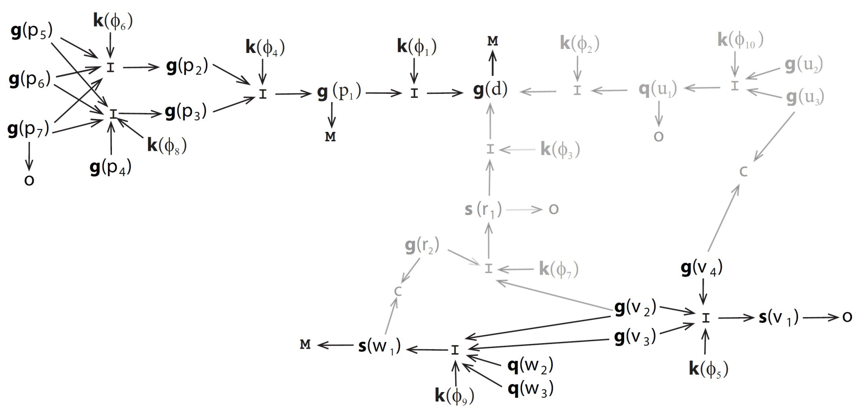
At this RE step, the problem faced by the requirements engineers and the stakeholders is to choose among the six potential solutions obtained. Optima, the optimization model proposed in this paper, can help to make this kind of decisions.
Section 4.1 describes Optima and how it is built while Section 4.2 proposed a short methodology in order to use it.



(a) The candidate solution S_1 highlighted



(b) The candidate solution S_2 highlighted



(c) The candidate solution S_3 highlighted

Fig. 5 The highlighted candidate solutions of the r-net illustrated in Fig. 4

4.1 Construction of the optimization model

4.1.1 Model setup

For each task t_i leading to the implementation of a requirement in the system-to-be (further explanations concerning the notion of *task* are given in §5), its implementation costs IC_i are taken from the quote of the software company in charge of the system implementation. Or, if that quote is not available, IC_i must be estimated.

The estimated profit of a task t_i can be computed as follows:

$$r_i = -IC_i + FR_i - FC_i + n \times varR_i - n \times varC_i \quad (7)$$

where FR_i and FC_i are respectively the periodical fixed revenue and fixed costs associated to the implementation of the task t_i, n is the estimated number of purchases achieved, and $varR_i$ and $varC_i$ are respectively the variable revenue and the variable costs belonging to the requirement i implemented through t_i. They can be computed by analyzing the technical documentation of the WS providers and by carrying out market researches. The pricing information is very often included in these WS level agreements. The reader can refer to [10] and [17] for more information about specific estimation techniques. At least, FC_i takes into account the maintenance of the part of the system-to-be that is associated to t_i. FR_i and FC_i also depend on the business of the services intermediary as well as on the objective(s) met by task t_i. They have to take into account all the fixed costs and the fixed revenue faced by the company when the task t_i is part of the system-to-be.

4.1.2 Estimation of the number of purchases

Seeing that the system-to-be of the services intermediary is assumed to be built from scratch, we do not have any data about previous sales and purchases. Thus, we cannot use existing models based on empirical data which enable to predict the future purchase behaviours of the customers and all the financial information needed to make decisions about the design of the system-to-be. Nevertheless, it is often said that opening a new website is similar to opening a new store located in a very dark cul-de-sac where nobody goes. This means that the services intermediary will very likely launch marketing campaign on the Internet. The most common alternative to older Internet-based advertising methods such as banners and pop-ups is the paid search advertising [26]: companies bid on some keywords from research engines such as Yahoo!, Bing and Google. Note that a keyword may consist of several individual words. The results obtained with paid search is much more interesting than other types of Internet-based ads for the companies buying the keywords [26]. The companies pay search engines for each click on the keywords bought. As explained by Laffey [26], one of the main advantages of paid search advertising is the possibility of targeting the audience of the ads. Readers can refer to the work of Lee & Seda [28] in order to have more information about paid search advertising.

In order to compute n (see Equation 7) we use the tool AdWords of Google as well as a model enabling to estimate the conversion rate. This model has been proposed by Ghose & Yang [14, 15]. It allows to estimate the conversion rate q for a keyword k at week w (see Equation 9 and related explanations). The choice of Google is based on the market share of this company (around 65%) compared with the market share of Yahoo! (around 19%) and Bing (around 9%) [46].

Estimating the number of clicks. The services intermediary has to choose a list of keywords on which it would like to make auctions as well as a maximum bid per auction. The services intermediary might also give a maximum daily budget. Thanks to AdWords, the daily number of visits (click) on the Web site is estimated. This is the daily number of clicks. The following ten keywords are the ones used in Adwords for USAtrip: *hotel New-York, travel to New-York, tour* USA*, New-York tour,* USA*trip, airlines to* USA*, travel in* USA*, Hilton New-York, cheapest flight to New-York* and *route planner* USA. The maximum bid for a click is fixed at € 0.82 thanks to the advice and the methodology given in [28]. The fixed budget, whatever the implemented solution will be, is limited to € 60 a day. This decision is optional: Google Adwords allows to place bids on different keywords without any daily limits. However, it is highly recommended to control and restrict this kind of marketing budget [28]. The location chosen is "The United Kingdom and Ireland" and the language is "English" because of the targeted market of the USAtrip company.

Estimating the conversion rate. For each chosen keyword, we use a vector K based on its wordographics. This concept is used in the literature studying the performance of keywords in paid search ads: the wordographics are the relevant characteristics of a keyword that can be used to estimate its position in a list of keywords, its click-through rate as well as its conversion rate [42]. Here are the wordographics used:

$$K_i = [NberClick, Rank, Retailer, Brand, Length] \quad (8)$$

where $NberClick$ is the weekly number of clicks, $Rank$ is the rank of the keyword i when it is displayed (both are given by Google AdWords). $Retailer$ and $Brand$ are binary variables indicating, respectively, if the keyword i contains a word related to the vendor name or to a well-known brand. $Length$ is the number of words in the keyword i.

To estimate the conversion rate, we need an additional variable: $PageQuality$. It captures the page quality behind the keyword hyperlink. The $PageQuality$ variable is computed by Google. Seeing that the underlying algorithm is not publicly available, we have to estimate this variable based on the information provided by Google[2]. The three main characteristics analyzed by the Google algorithm are the following:

[2] See the explanations provided by Google at: http://adwords.google.com/support/aw/?hl=fr (Last consultation on November 2012, the 5^{th}).

- The relevance and originality capture the quality of the content proposed by the Web site. A substantial and unique amount of information increases the Landing Page Quality computed by Google.
- The transparency is related to the clarity of the Web site. It is composed of three sub-characteristics: (i) the nature of the business done through the Web site is unambiguous, (ii) the interactions achieved with the browser are not hidden to the visitor and (iii) the amount of personal information asked to the visitor before registration is low and the privacy policy is well respected.
- The navigability represents the navigation ease. Web site visitors should easily find what they look for. Non-related and useless contents are drawbacks while a clear and short navigation path is an advantage when Google compute the Landing Page Quality.

The Landing Page Quality depends on the solution S_j and not on the specific keyword based on which the visitor arrives on the Web site. Indeed, we assume that the hyperlink behind the chosen keywords is the same for each of them. Q_j captures the value of the $PageQuality$ for the solution S_j. It ranges from 1 (the lowest possible score) to 10 (the highest possible score). As proposed by Ghose & Yang [15], three experts rated the six possible system-to-be and provided the following ranking: the page quality for S_1 to S_6 are respectively 4, 5, 8, 7, 4 and 6.

The conversion rate for the keyword k used in the context of the solution j at week w is computed as follows[3] (adapted from [15]):

$$\begin{aligned}q_{kwj}=&exp(\theta_{k0}+\theta_{k1}\times Rank_k+1.123\times Retailer_k-\\&0.879\times Brand_k+0.152\times PageQuality_j+\\&0.067\times Time_{kw})\times(1+exp(\theta_{k0}+\theta_{k1}\times Rank_k+\\&1.123\times Retailer_k-0.879\times Brand_k+0.152\times\\&PageQuality_j+0.067\times Time_{kw}))^{-1}\end{aligned} \tag{9}$$

$$where: \theta_{k0}=-4.457+\varsigma^{\theta}_{k0}, \theta_{k1}=-0.282+\varsigma^{\theta}_{k1}$$
$$\begin{bmatrix}\varsigma^{\theta}_{k0}\\ \varsigma^{\theta}_{k1}\end{bmatrix}\sim MVN\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1.426 & -0.131\\ -0.131 & 0.058\end{bmatrix}\right)$$

θ_{k0} and θ_{k1} capture the unobserved heterogeneity of the model. They vary along their respective means ($\overline{\theta_0}=-4.457$ and $\overline{\theta_1}=-0.282$). The variables $Rank_k$, $Retailer_k$ and $Brand_k$ come from the wordographics of each keyword. $Time_{kw}$ depends on the number of weeks between the begin of the advertising campaign and the week when q_{kwj} is estimated. $Time_{kw}$ is 1 for the first week and is then incremented each week.

[3] The initial model expects that the rank of each keyword varies each week [15]. Given that we use the rank number of Google that we considered as fixed over time, the $Rank$ variable only depends on the keyword.

Estimating the number of purchases. The number of purchases achieved is n_{jW}. It depends on the candidate solution S_j and on the number of weeks W. It can be computed as follows:

$$n_{jW}=\sum_{k=1}^{K}\sum_{w=1}^{W} nberClick_k\times q_{kwj} \tag{10}$$

where W is the number of weeks during which the software will probably be used and $nberClick_k$ is the weekly number of clicks on keyword k. q_{kwj} comes from the keywords analysis providing the conversion rate (see Equation 9).

4.1.3 The optimization model: Optima

Equation 7 is modified as follows: the profit r_i becomes the profit r_{ijW}. The latter is obtained from the implementation of the task t_i when the solution S_j is chosen for a given number of weeks W:

$$\begin{aligned}r_{ijW}=-IC_i+\frac{FR_i\times W}{Period}-\frac{FC_i\times W}{Period}+\\ n_{jW}\times varR_i-n_{jW}\times varC_i\end{aligned} \tag{11}$$

where $Period$ is the period for which the fixed revenue FR_i and the fixed costs FC_i are valid. The fixed costs and the fixed revenue can be paid or cashed each month, each semester, each year, etc. $Period$ is always expressed in weeks.

Equation 11 could be adapted to take into account the amortization of the implementation costs IC_i. This alternative is as follows:

$$\begin{aligned}r_{ijW}=\frac{-IC_i\times W}{Amort}+\frac{FR_i\times W}{Period}-\frac{FC_i\times W}{Period}+\\ n_{jW}\times varR_i-n_{jW}\times varC_i\end{aligned} \tag{12}$$

where the variable $Amort$ is the number of weeks needed for the whole amortization.

A subset of the requirements set φ_j –it comes from the r-net– is associated to each potential solution S_j. The tasks to implement in order to build the solution S_j are:

$$\bigwedge_i t_i \ \ \forall i\in\varphi_j\rightarrow S_j \tag{13}$$

For a given W, the estimated profit obtained from the implementation of S_j is:

$$R_{jW}=\sum_{i\ \in\varphi_j} r_{iW} \tag{14}$$

Once all the R_{jW} values are computed, the ranking of the potential solutions allow the requirements engineer(s) and the stakeholders to compare the solution from a financial point of view. Note that the advertising costs paid to the search engine company are not taken into account in Optima. They are indeed fixed costs without any influence on the ranking.

4.2 The use of Optima into six steps

We propose here a methodology in which we explain the process to follow in order to use Optima. Taking back Fig. 2 may help the reader understand this methodology. The six steps to follow are:

1. Build the r-net model based on the requirements elicitation work.
2. Manually or automatically [12] identify the candidate solutions as well as the minimum sub r-net containing all the mandatory requirements.
3. Extend the set of potential solutions by removing one or several optional requirement(s) from one of the potential solutions identified in Step 2. The lightened potential solution becomes a new element of the extended set of potential solutions.
4. Apply step 3 for each potential solution identified at step 2 until all the new potential solutions correspond to the minimal sub r-net.
5. Use the optimization model expressed in Equation 14 (along with Equation 11 or Equation 12 depending on your needs) in order to compute the estimated profit obtained from each solution.
6. Rank in descending order the solutions based on their respective estimated profits.

Optima has been implemented as a MatLab© function. The latter is available by sending an email to the first author.

5 Illustration of the use of Optima through an example

In this section, we use Optima during the USAtrip project. The extended set of potential solutions is composed of six different solutions[4] summarized in Table 2.

Table 2 Solutions of the Techne r-net presented in Fig. 4

	$g(p_7)$	$q(u_1)$	$s(v_1)$
S_1	No	No	No
S_2	Yes	Yes	No
S_3	Yes	No	Yes
S_4	Yes	No	No
S_5	No	Yes	No
S_6	No	No	Yes

[4] Note that the requirements listed in the first row of Table 2 are all optional requirements. See §6 in which a short comment is made about requirements capturing stakeholders' preferences.

First, we add the tasks m_1 to m_{11} to the r-net depicted in Fig. 4. This is illustrated in Fig. 6. Those tasks capture the specifications of the software pieces which are linked to the related goal or quality constraints. We only instantiate the goals and quality constraints contained in at least one of the six potential solutions. That is why the softgoal $s(r_1)$ and its refinement is no longer part of the current discussion.

For each task which is the leaf of an optional antecedent ($t(m_4), t(m_5), t(m_6)$ and $t(m_8)$), an evaluation of the implementation costs, the fixed revenue and costs, and the variable revenue and costs is achieved. The other tasks are not included in our optimization model since they are part of all solutions. Indeed, they cannot be a source of revenue variability. They are thus irrelevant in an optimization problem issue.

The implementation costs are based on the estimate of the company that will implement the system-to-be. Concerning the estimation of the fixed and variable revenue, they can be either based on market research or on the future price that the company plans to apply. For each task $t(m_i)$ taken into account in Optima, a vector t_i is computed. All the vectors t_i's put together form a costs and revenue matrix used in the MatLab function.

$$t_i = [IC_i, FR_i, FC_i, varR_i, varC_i] \tag{15}$$

We achieve this work for $t(m_4), t(m_5), t(m_6)$ and $t(m_8)$. They are the four requirements on which an implementation decision has to be made: *From a financial point of view, is it relevant to implement them in the system-to-be?*. For instance, $t(m_4)$ is associated to the following vector: $[17000, 0, 20800, 96, 82]$. We use the optimization model proposed in Equation 14 with an estimation period of 60 weeks. This is the expected period of time in which the future solution should be used by USAtrip without any significant changes in the IS.

The results of the simulation give the solution S_5 as the best system-to-be to implement. The ranking provided by Optima is transcribed in Table 3.

Table 3 Ranking of the six potential solutions along with their expected profits

	Solution	Expected profit
1	S_5	€ 146 900
2	S_2	€ 122 130
3	S_4	€ 38 820
4	S_3	€ 16 170
5	S_1	€ 0
6	S_6	€ −2 600

Based on the simulation, we can advise the USAtrip company to implement S_5: this solution should be the most profitable. The figures issued from the simulation are not absolute

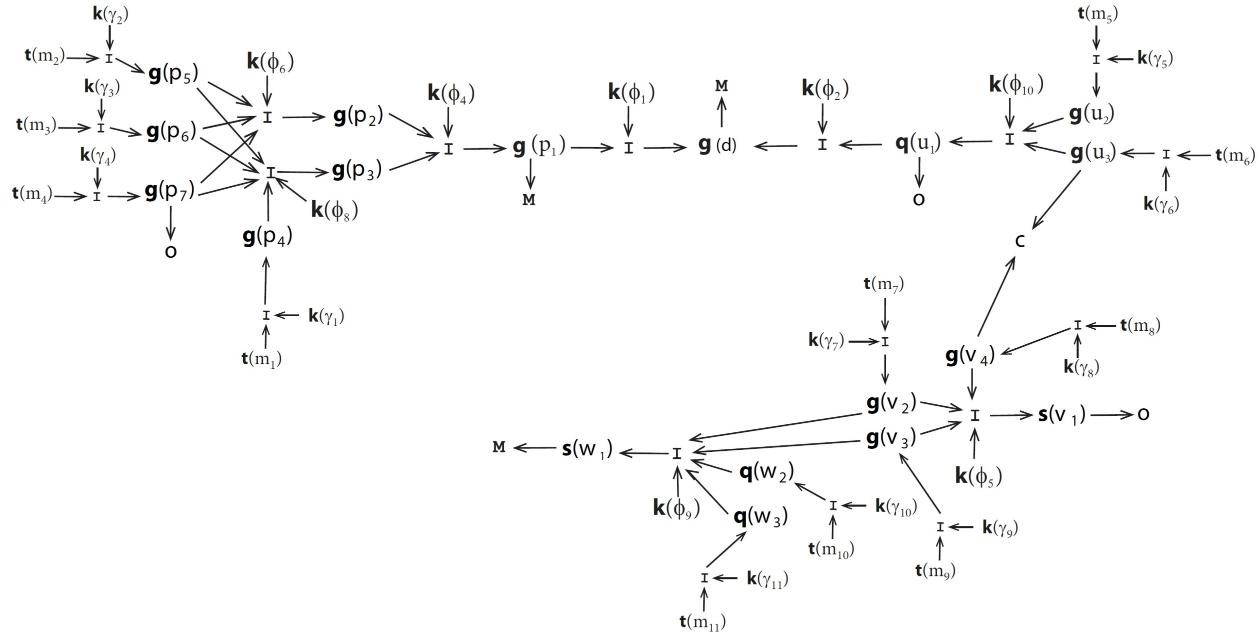


Fig. 6 The r-net of the USAtrip company along with the implementation Tasks

although hundreds of simulations based on the same input data are highly convergent: the average standard deviation is about 10% of the mean (see §5.1).

Note that the expected profits will not be the net profits of the company. Indeed, we only take into account the costs and revenue linked to the optional and conflicting requirements of the system-to-be. The figures listed in Table 3 have to be added to the costs and revenue obtained from the whole system-to-be. Furthermore, the company could also face costs and may obtain revenue which are not directly linked to the implementation of the IS designed. Nevertheless, Optima provides relevant financial information in order to make decisions about the optional and conflicting requirements. This information can be helpful during the subsequent phases of the software engineering process which aims at specifying the single solution that will be implemented.

5.1 Analysis of the results coming from multiple simulations

We achieved thousand simulations based on the described case example. To do this, we use the MatLab function implementing the optimization model developed in this paper. In Fig. 7, a scatter plot shows the dispersion of the expected profits for each of the simulations. Horizontal lines are the means; their values are: $\overline{S_1} = 0$, $\overline{S_2} = 118\ 480$, $\overline{S_3} = 15\ 461$, $\overline{S_4} = 36\ 997$, $\overline{S_5} = 148\ 998$ and $\overline{S_6} = -2\ 600$. The significant standard deviations are $\sigma_{S_2} = 9\ 597$, $\sigma_{S_2} = 4\ 032$, $\sigma_{S_2} = 4\ 566$, $\sigma_{S_2} = 11\ 027$. The standard deviation of S_1 and S_6 is zero.

Among the thousand simulations, the ranking of the solutions stated in Table 3 is valid 988 times. For the few other cases, Solutions 2 and 5 are reversed. Once two solutions are significantly close to each other, the rankings got from multiple simulations will differ. At the same time, the means of their expected profits will also be closer. This situation should be interpreted as follows: from a financial viewpoint, there is (almost) no difference between these two potential solutions. In order to choose one or the other solution, stakeholders and requirements engineers are asked to use the preferences elicited during the RE stage (see §6) or other aspects of decision-making in prioritization and conflicts resolution. Some of these other aspects are briefly tackled in the related work (§7).

6 The management of the preferences expressed by stakeholders

During the RE stage, some requirements elicited captured preferences expressed by the stakeholders. Those preferences can either assess positively or negatively an individual requirement, or rank two different requirements. For instance, two goals could be ranked or a quality constraint can be individually appraised.

Techne also provides a constructor to design those preferences in an r-net (see Fig. 3(d)). So far, the discussion and the proposed optimization model ignore this fact: we work with an r-net, free of any preference. The result provided by Optima is a recommendation which aims at informing the stakeholders, the requirements engineers as well as the decision makers about the likely consequences of their future choices. This recommendation is only financial, against which other aspects could be opposed (e.g., ethical values

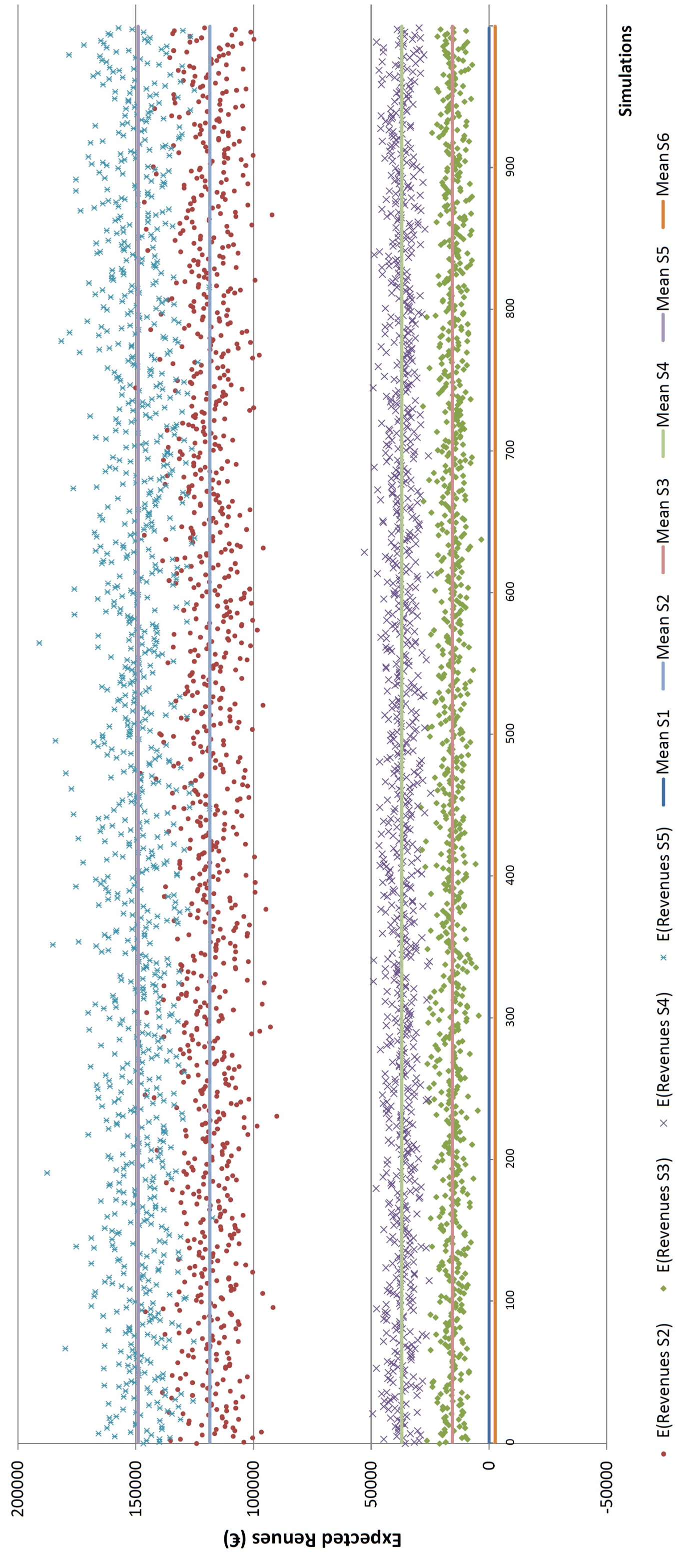


Fig. 7 Plot of the expected profits obtained from the candidate solutions (thousand simulations)

of some stakeholders). If a preference differs from the financial recommendation coming from our optimization model, the decision makers have to make their own decisions. They could go against the results issued from Optima. Providing some guidelines or a process to follow in order to solve those

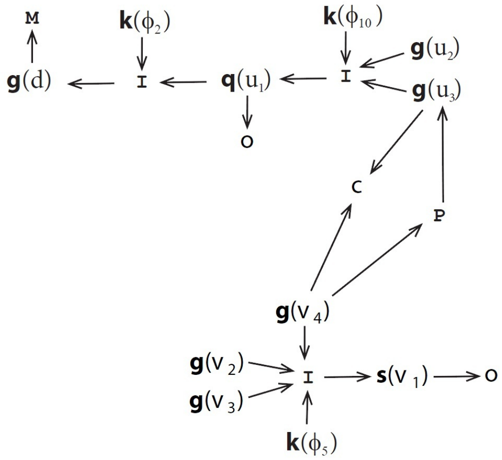


Fig. 8 Example of a preference opposing the financial recommendation of Optima

kinds of conflicts is out of the scope of this paper. It is left for future work.

6.1 Preferences in the requirements model of USAtrip

To illustrate how we advise to manage the preferences when Optima is used, we provide an extract of Fig. 4 in which a stakeholder expresses a strong preference. In this case, the marketing director prefers the use of an identical layout when the travel offers are displayed compared with the highlighting of the sponsored travel offers.

In Fig. 8, you can notice that the requirement $\mathbf{g}(v_4)$ is strictly preferred to the requirement $\mathbf{g}(u_3)$. Therefore, each solution having $\mathbf{g}(v_4)$ instead $\mathbf{g}(u_3)$ as the only difference is preferred by the marketing director. That means that solution S_6 is strictly preferred to S_5. However, *from a financial point of view*, this preference clearly contradicts the recommendations coming from the use of Optima. S_5 is indeed rated as the best solution while S_6 is rated as the worst solution. In this case, a choice has to be made between the preference of the marketing director and the financial advantage derived from the implementation of the solution S_5.

7 Related work

This work is mainly related to two sub-domains of RE: the prioritization of requirements and the resolution of conflicts between requirements. These two issues are considered as very significant in many seminal works [9, 27, 34]. Nevertheless, these two challenges are often put aside in lots of works about analysis and reasoning on requirements in Web-based environments such as in [11, 31].

7.1 Requirements prioritization

Requirements prioritization is often considered as a complex task in RE. Quite simple as well as more difficult techniques exist [50]. There are two main kinds of approaches to tackle this issue: the negotiation approaches and the quantitative approaches. The negotiation approaches are based on subjective measures. In this case, the prioritization work achieved is mostly based on textual documents and discussions between stakeholders. This approach is recommended when the decision variables are highly interrelated.
The method proposed in this paper falls into the second category. The quantitative approaches assign values to the decision variables. The requirement with the higher aggregated results is ranked first, and so on. The variables taken into account are, mainly, (i) penalties when a requirement is not implemented, (ii) implementation and maintenance costs, (iii) the time needed to implement a requirement, (iv) internal and external risks, (v) the volatility of a requirement, (vi) the financial benefit obtained from a requirement, and (vii) the ease of implementation [4, 50]. As other techniques and methods [24, 51], the optimization model proposed in this work combines several aspects: the cost of implementation and maintenance, and the financial benefit (penalty) when a requirement is (or is not) implemented. What aspects should be considered mainly depends on the specific environment and the stakeholders of the system-to-be.

The Analytical Hierarchy Process (AHP) [43] (a.k.a. the pair-wise comparison technique) is a well-known technique. However its use is difficult for large numbers of requirements [32]. In our specific environment, the AHP method is unsatisfactory since the amount of variables to handle is too large for human beings. Moreover, the prioritization is often based on other aspects than the ones used in our method.

The Cost-value Approach [24] roughly takes into account the same aspect as us, i.e., the value obtained from the implementation of a given requirement and its costs of implementation. The Cost-value Approach is a pairwise comparison of the optional requirements based on the AHP method. It mainly differs in terms of scope, ease of use and processes followed. Optima allows to objectify the profit received from the implementation of a given requirement while, in the Cost-value Approach, the value is relative and stated by stakeholders. Moreover, we take into account the interdependencies between requirements: we indeed compare complete potential solutions and not some single requirements as it is the case in the Cost-value Approach.

There are also tools that allow to analyse requirements in order to find the best solutions. These mainly focus on the non-functional requirements. Up to now, the most comprehensive and performing tool seems to be the Evolutionary Requirements Analyzer [45].

7.2 Resolution of conflicts in requirements models

Some authors also investigate the resolution of conflicts in RE. In [50], van Lamsweerde defines conflict as "strong inconsistencies [. . .] that cannot be satisfied when taken together". They are often resolved with discussions and negotiations between stakeholders. Several techniques and tools exist. The most representative solutions are the iterative resolution [38, 41], the problem restructuring [25] or the use of stakeholders' utility functions [19]. They always rely on discussions and negotiations or ask for a modification of the requirements model [39]. Other initiatives such as [33] propose to explore alternatives during RE but they do not really offer a technique or a method to choose among the possible alternatives.

In this paper, the optimization model proposed takes into account all the potential solutions to the requirements problem –thanks to Techne and the underlying requirements ontology [22]– and ranks them in order of profitability. The advantage of using Optima is that the negotiation can be (partially) based on objective information instead of on the stakeholders' preferences which are (too) often subjective [44].

7.3 Recommender systems

The last related works concern the recommender systems [1, 13]. Clearly, Optima could be a relevant tool to include in these systems. It might allow their use for the creation of ISs for services intermediaries. Nevertheless, within this perspective, it is important to keep in mind that "*requirements prioritization practices are informal and dependent of individuals. [That means that] having systematic requirements prioritization practices is a challenge because requirements prioritization requires a great deal of non-trivial decision making*" [29]. This is a reason why we would rather provide a model to use as an indicator useful for the decision-making process instead of a comprehensive and rigid tool leading to very few contacts between stakeholders, requirements engineers and developers.

8 Conclusion and future works

The re-intermediation of supply chains based on ICT in service-oriented environments raises new issues. One of them lies in the decision-making process when multiple solutions exist in the requirements model: *How to determine which potential solution must be implemented?*

The optimization model proposed in this paper (Optima) provides an answer from a financial viewpoint. It should be used on a paraconsistent requirements model. That kind of models accepts conflicts when one reasons over them. At once, Optima resolves the conflicts between requirements and prioritizes the optional requirements. As a result, it gives a ranking of the potential solutions identified in the requirements model. This ranking is a very interesting basis for further discussions and negotiations, or it makes the decision.

Because Optima is partially based on a stochastic equation, we advise to make estimations on several weeks, and make several estimations with the same data. Unchanged results show that the different solutions are strongly different from a financial standpoint, while dissimilar rankings point out that the solutions are not so different from a financial point of view. In all cases, the decision to implement one or the other solution can take into account other aspects of the decision which are stated as important by the stakeholders. Nevertheless, all of our tests show that the proposed optimization model gives a convergent solution once the number of estimations increases.

The main future work lies in the study of the decision-making process leading to IS modifications. Once the initial system is used every day, the question is *When to change the fundamental structure of the system in use?*

A second work to achieve is to manage and, if possible, to integrate in Optima the preferences expressed by the stakeholders and structured in the requirements model. Currently, stakeholders preferences are not part of the discussion although they could contradict the results provided by Optima.

Lastly, the current version of Optima takes into account the financial aspects of the problem tackled. There are other important aspects to take into account when choosing the future IS of a services intermediary among several solutions. We can mention, for instance, the time needed for their implementation or the increase/reduction of risk (to define!) when a requirement is implemented.

References

1. Alenljung B, Persson A (2006) Decision-Making Activities in Requirements Engineering Decision Processes: A Case Study. In: Advances in Information Systems Development: Bridging the Gap between Academia and Industry, Springer, pp 707–718
2. Bakos Y (1998) The Emerging Role of Electronic Marketplaces on the Internet. Commun of the ACM 41(8):35–42
3. Bano M, Ikram N (2010) Issues and Challenges of Requirement Engineering in Service Oriented Software Development. In: The Fifth International Conference on Software Engineering Advances (ICSEA 2010), IEEE Computer Society, pp 64–69
4. Berander P, Andrews A (2005) Requirements prioritization. In: Aurum A, Wohlin C (eds) Engineering and Managing Software Requirements, Springer, chap 4, pp 69–94

5. Bieberstein N, Bose S, Fiammante M, Jones K, Shah R (2006) Service-Oriented Architecture (SOA) Compass: Business Value, Planning, and Enterprise Roadmap. Prentice-Hall
6. Boyle E, Humphreys P, McIvor R (2008) Reducing supply chain environmental uncertainty through e-intermediation: An organisation theory perspective. Int J of Prod Econ 114(1):347–362
7. Casati F, Ilnicki S, Jin Lj, Krishnamoorthy V, Shan MC (2000) Adaptive and Dynamic Service Composition in eFlow. In: Proceedings of the 12th International Conference on Advanced Information Systems Engineering (CAISE '00), Springer-Verlag, pp 13–31
8. Chaffey D (2007) E-Business and E-Commerce Management, 3rd edn. Prentice-Hall
9. Cheng BH, Atlee JM (2009) Current and Future Research Directions in Requirements Engineering. In: Lyytinen K, Loucopoulos P, Mylopoulos J, Robinson B (eds) Design Requirements Engineering: A Ten-Year Perspective, Lecture Notes in Business Information Processing, vol 14, Springer-Verlag Berlin Heidelberg, pp 11–43
10. Churchill GA, Iacobucci D (2009) Marketing research: methodological foundations, 10th edn. South-Western College Pub.
11. Cuaresma MJE, Koch N (2004) Requirements engineering for web applications - a comparative study. J of Web Eng 2(3):193–212
12. Ernst NA, Mylopoulos J, Borgida A, Jureta I (2010) Reasoning with optional and preferred requirements. In: 29th International Conference on Conceptual Modeling (ER 2010), Springer, Lecture Notes in Computer Science, vol 6412, pp 118–131
13. Felfernig A, Schubert M, Mandl M, Ricci F, Maalej W (2010) Recommendation and decision technologies for requirements engineering. In: Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering, ACM Press, pp 11–15
14. Ghose A, Yang S (2008) An Empirical Analysis of Sponsored Search Performance in Search Engine Advertising. In: Proceedings of the First International Conference on Web Search and Web Data Mining, ACM Press, pp 241–250
15. Ghose A, Yang S (2009) An Empirical Analysis of Search Engine Advertising: Sponsored Search in Electronic Markets. Manag Sci 55(10):1605–1622
16. Giaglis GM, Klein S, O'Keefe RM (2002) The role of intermediaries in electronic marketplaces: developing a contingency model. Inf Syst J 12(3):231–246
17. Gouscos D, Kalikakis M, Georgiadis P (2003) An approach to modeling Web service QoS and provision price. In: Proceedings of the Fourth International Conference on Web Information Systems Engineering, IEEE Computer Society, pp 121–130
18. Hill TP (1977) On goods and services. Rev of Income and Wealth 23(4):315–352
19. Jacobs S, Kethers S (1994) Improving Communication and Decision Making within Quality Function Deployment. In: Proceedings of the First Conference on Concurrent Engineering Research and Application, Concurrent Technologies Corporation, pp 203–213
20. Jelassi T, Enders A (2008) Strategies for E-Business: Creating Value Through Electronic and Mobile Commerce, 2nd edn. Prentice-Hall
21. Jureta IJ, Mylopoulos J, Faulkner S (2008) Revisiting the Core Ontology and Problem in Requirements Engineering. In: Proceedings of the 16th IEEE International Requirements Engineering Conference (RE 2008), IEEE Computer Society, pp 71–80
22. Jureta IJ, Borgida A, Ernst NA, Mylopoulos J (2010) Techne: Towards a New Generation of Requirements Modeling Languages with Goals, Preferences, and Inconsistency Handling. In: Proceedings of the 18th IEEE International Requirements Engineering Conference (RE 2010), IEEE Computer Society, pp 115–124
23. Kalakota R, Whinston AB (1997) Electronic Commerce: A Manager's Guide. Addison-Wesley Longman Publishing Co., Inc.
24. Karlsson J, Ryan K (1997) A Cost-Value Approach for Prioritizing Requirements. IEEE Softw 14(5):67–74
25. Klein M (1991) Supporting conflict resolution in cooperative design systems. IEEE Trans on Syst, Man and Cybern 21(6):1379–1390
26. Laffey D (2007) Paid search: The innovation that changed the Web. Bus Horiz 50(3):211–218
27. van Lamsweerde A (2000) Requirements Engineering in the Year 00: A Research Perspective. In: Proceedings of the 22nd International Conference on Software Engineering (ICSE'2000), ACM Press, pp 5–19
28. Lee K, Seda C (2009) Search Engine Advertising: Buying Your Way to the Top to Increase Sales, 2nd edn. New Riders Press
29. Lehtola L, Kauppinen M, Kujala S (2004) Requirements Prioritization Challenges in Practice. In: Bomarius F, Iida H (eds) Product Focused Software Process Improvement, Springer, Lecture Notes in Computer Science, vol 3009, pp 497–508
30. Lo A, Yu E (2007) From Business Models to Service-Oriented Design: A Reference Catalog Approach. In: Proceedings of the 26th International Conference on Conceptual Modeling (ER 2007), Springer, Lecture Notes in Computer Science, vol 4801, pp 87–101
31. Luna ER, Garrigós I, Grigera J, Winckler M (2010) Capture and Evolution of Web Requirements Using WebSpec. In: Proceedings of the 10th International Conference on Web Engineering (ICWE 2010), Springer, Lec-

ture Notes in Computer Science, vol 6189, pp 173–188
32. Maiden NAM, Ncube C (1998) Acquiring COTS Software Selection Requirements. IEEE Softw 15(2):46–56
33. Mylopoulos J, Chung L, Liao S, Wang H, Yu ESK (2001) Exploring Alternatives during Requirements Analysis. IEEE Softw 18(1):92–96
34. Nuseibeh B, Easterbrook SM (2000) Requirements Engineering: A Roadmap. In: Proceedings of the Conference on the Future of Software Engineering (ICSE 2000) - Future of SE Track, ACM Press, pp 35–46
35. Osterwalder A, Pigneur Y (2010) Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers, 1st edn. Wiley
36. Osterwalder A, Pigneur Y, Tucci CL (2005) Clarifying business models: Origins, present, and future of the concept. Commun of the Assoc for Inf Syst 16(1):1–25
37. Papazoglou MP, Georgakopoulos D (2003) Service-oriented Computing. Commun of the ACM 46(10):24–28
38. Robinson WN, Volkov V (1998) Supporting the negotiation life cycle. Commun of the ACM 41(5):95–102
39. Robinson WN, Pawlowski SD, Volkov V (2003) Requirements Interaction Management. ACM Comput Surv 35(2):132–190
40. Roy J, Ramanujan A (2001) Understanding web services. IT Prof 3(6):69–73
41. Ruhe G, Eberlein A, Pfahl D (2002) Quantitative Win-Win: A New Method for Decision Support in Requirements Negotiation. In: Proceedings of the International Conference of Software Engineering and Knowledge Engineering (SEKE 2002), pp 159–166
42. Rutz OJ, Bucklin RE (2007) A Model of Individual Keyword Performance in Paid Search Advertising. Tech. rep., Yale University
43. Saaty T (1980) The Analytic Hierarchy Process, Planning, Piority Setting, Resource Allocation. McGraw-Hill, New york
44. Strigini L (1996) Limiting the Dangers of Intuitive Decision Making. IEEE Softw 13(1):101–103
45. Sutcliffe AG, Chang WC, Neville R (2003) Evolutionary Requirements Analysis. In: Proceedings of the 11th IEEE International Conference on Requirements Engineering (RE 2003), IEEE Computer Society, pp 264–73
46. The UK Search Engine Marketing Agency (Browser Media) (September 24, 2009) Bing's slowly gaining on Google's Market Share. Available at: http://www.browsermedia.co.uk/2009/09/24/bings-slowly-gaining-on-googles-market-share
47. Tsai WT (2005) Service-oriented system engineering: A new paradigm. In: Proceedings of the IEEE International Workshop on Service-Oriented System Engineering (SOSE), IEEE Computer Society, pp 3–6
48. Tsai WT, Jin Z, Wang P, Wu B (2007) Requirement Engineering in Service-Oriented System Engineering. In: Proceedings of ICEBE 2007, IEEE International Conference on e-Business Engineering and the Workshops SOAIC 2007, SOSE 2007, SOKM 2007, IEEE Computer Society, pp 661–668
49. van Eck P, Wieringa R (2003) Requirements Engineering for Service-Oriented Computing: A Position Paper. In: First International Workshop on e-Services at The 5th International Conference on Electronic Commerce (ICEC'03), TU Delft, pp 23–28
50. van Lamsweerde A (2009) Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley
51. Wiegers K (2003) Software requirements. Microsoft Press